IN THE UNITED STATES DISTRICT COURT
FOR THE SOUTHERN DISTRICT OF TEXAS
GALVESTON DIVISION

Autoscribe Corporation	§	
	§	
Plaintiff,	§	
	§	
v.	§	Case No. _______________
	§	
Paysafe Limited, Paysafe Holdings (US) Corp., Paysafe Merchant Services Corp., and Paysafe Payment Processing Solutions LLC	§ § § §	JURY TRIAL DEMANDED
	§	
Defendants.		

PLAINTIFF'S ORIGINAL COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Autoscribe Corporation ("Autoscribe" or "Plaintiff") hereby submits this Original Complaint for patent infringement against Paysafe Limited, Paysafe Holdings (US) Corp., Paysafe Merchant Services Corp., and Paysafe Payment Processing Solutions LLC (together, "Defendants") and alleges, based on its own personal knowledge with respect to its own actions and based upon information and belief with respect to all others' actions, as follows:

I. THE PARTIES

1. Autoscribe is a Corporation organized under the laws of the state of Maryland with its principal place of business at 12276 San Jose Blvd, Suite 624, Jacksonville, FL 32223.

2. Paysafe Limited is a limited company organized under the laws of Bermuda.[1] Paysafe Limited can be served through its "company contact person": Elliott Wiseman, 2 Gresham Street, London, United Kingdom EC2V 7AD.[2]

3. Paysafe Holdings (US) Corp. is a corporation organized under the laws of

[1] https://ir.paysafe.com/financial-info-and-filings/sec-filings/content/0000950170-24-033944/psfe-20231231.htm (last visited March 31, 2024).

[2] *Id.*

Delaware, with regular and established places of business at 128 Vision Park Boulevard, Shenandoah, TX 77384 and 1725 Hughes Landing Blvd., 11th Floor, The Woodlands, TX 77380.[3] Paysafe Holdings (US) Corp. can be served through its registered agent: C-T Corporation, 1999 Bryan Street Suite 900, Dallas, TX 75201.

4. Paysafe Merchant Services Corp. is a corporation organized under the laws of Delaware, with a regular and established place of business at 128 Vision Park Boulevard, Shenandoah, TX 77384.[4] Paysafe Merchant Services Corp. can be served through its registered agent C-T Corporation, 1999 Bryan Street Suite 900, Dallas, TX 75201.

5. Paysafe Payment Processing Solutions LLC is a limited liability company organized under the laws of Delaware, with a principal place of business at 1725 Hughes Landing Blvd., 11th Floor, The Woodlands, TX 77380.[5] Paysafe Holdings (US) Corp. is a member of Paysafe Payment Processing Solutions LLC.[6] Paysafe Payment Processing Solutions LLC can be served through its registered agent: C-T Corporation, 1999 Bryan Street Suite 900, Dallas, TX 75201.

[3] Exhibit A (Texas Secretary of State Business Entity Search for Paysafe Holdings (US) Corp.); https://search.sunbiz.org/Inquiry/CorporationSearch/SearchResultDetail?inquirytype=EntityName&directionType=Initial&searchNameOrder=PAYSAFEPAYMENTPROCESSINGSOLUTI%20M170000051020&aggregateId=forl-m17000005102-67853c86-2058-4c5e-9f0b-8edc09eddd9c&searchTerm=paysafe&listNameOrder=PAYSAFECARDCOMUSA%20F210000028370.

[4] Exhibit B (Texas Secretary of State Business Entity Search for Paysafe Merchants Services Corp.).

[5] https://search.sunbiz.org/Inquiry/CorporationSearch/SearchResultDetail?inquirytype=EntityName&directionType=Initial&searchNameOrder=PAYSAFEPAYMENTPROCESSINGSOLUTI%20M170000051020&aggregateId=forl-m17000005102-67853c86-2058-4c5e-9f0b-8edc09eddd9c&searchTerm=paysafe&listNameOrder=PAYSAFECARDCOMUSA%20F210000028370.

[6] *Id.*

II. JURISDICTION AND VENUE

6. This Court has subject matter jurisdiction under 28 U.S.C. §§ 1331 and 1338(a) because this action arises under the patent laws of the United States, 35 U.S.C. § 1 *et seq.*, including 35 U.S.C. § 271.

7. As discussed in greater detail below, Defendants have committed acts of patent infringement and/or have induced and/or contributed to acts of patent infringement by others in this judicial district, the State of Texas, and elsewhere in the United States, by making, using, offering for sale, selling, or importing various products or services that infringe Autoscribe's Asserted Patent (defined below).

8. As mentioned above, Defendants Paysafe Holdings (US) Corp., Paysafe Merchant Services Corp., and Paysafe Payment Processing Solutions LLC have regular and established places of business at 128 Vision Park Blvd, Suite 300, Shenandoah TX 77384 and/or 1725 Hughes Landing Blvd., 11th Floor, The Woodlands, TX 77380, which are located in this district. Defendant Paysafe Payment Processing Solutions LLC lists the The Woodlands office as its principal address. Further, all Defendants are subject to personal jurisdiction in the Southern District of Texas because they have purposely availed themselves of jurisdiction in the state through their business in Texas, including but not limited to operations that contribute to the direct, indirect, or inducement of the infringing instrumentalities in the District.

9. Venue is proper in this judicial District under 28 U.S.C. § 1400(b) because Paysafe Payment Processing Solutions LLC resides in the District. Additionally, venue is proper as to Paysafe Holdings (US) Corp., Paysafe Merchant Services Corp., and Paysafe Processing Solutions LLC because these Defendants have committed patent infringement and/or has induced and/or contributed to acts of infringement by others in the District and have regular and established places of business in the District, as discussed above.

10. With regard to Paysafe, Ltd., venue is proper in this District under 28 U.S.C. § 1391(c) because Defendant Paysafe Ltd. is subject to this Court's personal jurisdiction and because, being an alien corporation, it may be sued in any district that has personal jurisdiction.

III. BACKGROUND

11. Fraud in credit card and other financial transactions is a major problem, particularly in the online marketplace. Considerable resources are devoted to securing credit card and other account information provided to online merchants by payers. A single breach of security incident can compromise millions of credit card accounts, and such breaches are reported on a regular basis. As such, customers' financial data are sensitive in nature and are subject to strict regulations. Companies that fail to adequately protect customers' credit card data may face significant legal and regulatory consequences.

12. Autoscribe is a leading financial services company and payment processor, currently processing more than $2 billion in transactions annually and servicing thousands of financial institutions and corporate billers across the nation. As part of its mission, Autoscribe has invested significant resources and capital into developing new technologies to facilitate transactions and assist billers in meeting their compliance needs while minimizing costs and complexity.

13. Autoscribe has protected these technologies with a robust and growing patent portfolio.

14. On April 4, 2023, the United States Patent and Trademark Office ("USPTO") duly and legally issued United States Patent No. 11,620,621 ("the '621 Patent" or "the Asserted Patent"), titled "Enrolling a payer by a merchant server operated by or for the benefit of a payee and processing a payment from the payer by a secure server." The Asserted Patent is valid and enforceable.

15. The Asserted Patent is directed to "systems and methods for obtaining and using account information to process financial payments."

16. Autoscribe is the original applicant and the sole and exclusive owner of all rights, title, and interest in the Asserted Patent, including the sole and exclusive right to prosecute this action, to enforce the Asserted Patent against infringers, to collect damages for past, present and future infringement of the Asserted Patent, and to seek injunctive relief as appropriate under the law.

17. Autoscribe has complied with any marking requirements under 35 U.S.C. § 287 with regard to the Asserted Patent.

18. Defendants provide a "payments platform with an extensive track record of serving merchants and consumers in the global entertainment sectors," which enables "businesses and consumers to connect and transact seamlessly through" its "capabilities in payment processing, digital wallet, and online cash solutions."[7] Defendants had "an annualized transaction volume of over $130 billion in 2022."[8]

19. As discussed in greater detail below, Defendants provide and use processing solutions, including their "Paysafe JS" and "Payment API" products, that are covered by the Asserted Patent.

20. Defendants compete directly against Autoscribe, including through their "Paysafe JS" and "Payment API" products, causing Autoscribe to lose significant profits.

21. Accordingly, Defendants' infringement, as described below, has injured, and continues to injure Autoscribe.

[7] https://ir.paysafe.com/company-information (last visited March 26, 2024).
[8] *Id.*

IV. COUNT I: INFRINGEMENT OF THE ASSERTED PATENT

22. Autoscribe incorporates each of the allegations of paragraphs 1–21 above.

23. Defendants have directly infringed and continue to directly infringe the Asserted Patent by, for example, making, using, offering to sell, selling, and/or importing into the United States, without authority, products or services that practice one or more claims of the Asserted Patent.

24. Defendants are not licensed or otherwise authorized to make, use, offer for sale, sell or import any products or services that embody the inventions of the Asserted Patent in the United States.

25. Defendants have and continue to directly infringe one or more claims of the Asserted Patent, including, for example, claim 1, either literally or under the doctrine of equivalents, by performing every step of the claimed method in violation of 35 U.S.C. § 271.

26. Defendants' infringing services include, for example, the services Defendants provide through their "Paysafe JS" and "Payment API" products, as well as any other similar methods performed by Defendants (collectively, the "Infringing Methods").

27. For example, Representative Claim 1 of the Asserted Patent claims:

> A method of processing a payment transaction from a payer to a payee, the method being performed by one or more secure servers, the method comprising:
>
>> providing, by the one or more secure servers to a merchant server providing a webpage to a payer computing system used by the payer, an application programming interface (API) that:
>>
>> provides financial account registration and token retrieval functions that can be executed to process the payment transaction;
>>
>> provides access to the financial account registration and token retrieval functions to the merchant server;

receives, from the merchant server via the API, at least one data element associated with the payer and a payment amount from the payer to the payee;

authenticates the payee; and

executes the financial account registration function, upon initiation by the merchant server, by:

> generating a uniform resource locator (URL), for establishing a secure socket layer connection via the internet between the secure server and the payer computing system, the URL comprising either:
>
> > a dynamic URL generated by the secure server for the payer and the payee; or a static URL and a hypertext transport protocol (HTTP) parameter used by the secure server to identify the payer and the payee;
>
> establishing the secure socket layer connection, in response to an HTTP request received from the merchant server for the generated URL, between the secure server and the payer computing system within a window or frame that is displayed within the webpage provided by the merchant server;
>
> outputting instructions to the payer computing system, in response to the HTTP request for the generated URL, to render a financial account registration request form, within the window or frame that is displayed within the webpage provided by the merchant server, that provides functionality for the payer to provide sensitive financial account information associated with a financial account; and
>
> outputting instructions to the payer computing system, in response to the HTTP request for the generated URL, to encrypt the sensitive financial account information provided by the payer and transmit the encrypted financial account information to the secure server via the secure socket layer connection;

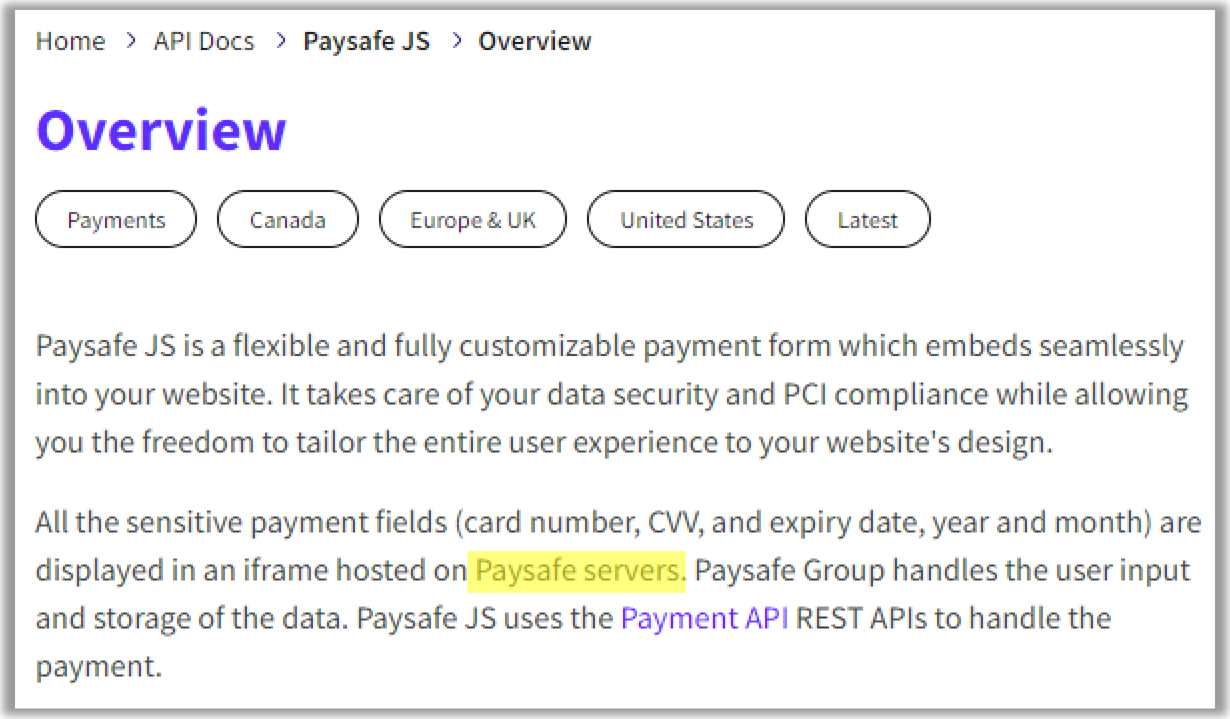
receiving the sensitive financial account information provided by the payer via the secure socket layer connection;

storing the sensitive financial account information in a secure storage location and performing each software process required to maintain compliance with one or more information security standards;

executing a token retrieval function, upon initiation by the merchant server via the API, by:

> providing a non-sensitive electronic data token representing the sensitive financial account information to the merchant server without providing the sensitive financial account information to the merchant server and without providing the non-sensitive electronic data token to the payer; and
>
> processing the payment transaction using the sensitive financial account information by generating and transmitting an electronic request requesting the payment amount from the financial account, obtaining the payment amount, and forwarding at least a portion of the payment amount to the payee.

28. Through their "Paysafe JS" and "Payment API" products, Defendants perform a method of processing a payment transaction from a payer to a payee, the method being performed by one or more secure servers and meeting every element of Claim 1. The figure below are excerpts from Defendants' documentation for their "Paysafe JS" and "Payment API" products:[9]

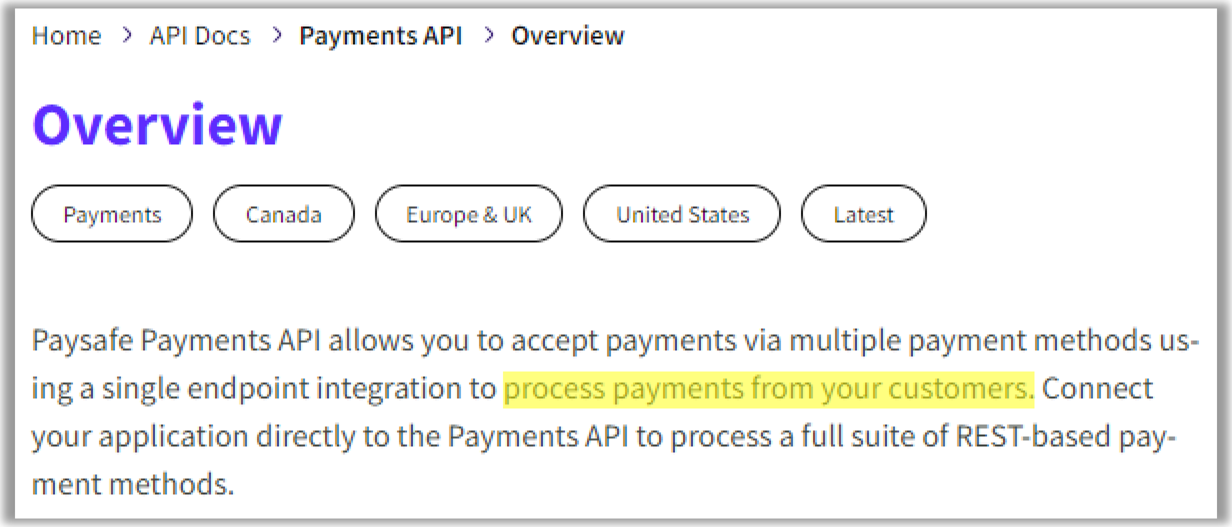


Overview

Paysafe JS is a flexible and fully customizable payment form which embeds seamlessly into your website. It takes care of your data security and PCI compliance while allowing you the freedom to tailor the entire user experience to your website's design.

All the sensitive payment fields (card number, CVV, and expiry date, year and month) are displayed in an iframe hosted on Paysafe servers. Paysafe Group handles the user input and storage of the data. Paysafe JS uses the Payment API REST APIs to handle the payment.

[9] https://developer.paysafe.com/en/api-docs/paysafe-js/overview/ (last visited March 31, 2024); https://developer.paysafe.com/en/api-docs/payments-api/overview/ (last visited March 31, 2024).



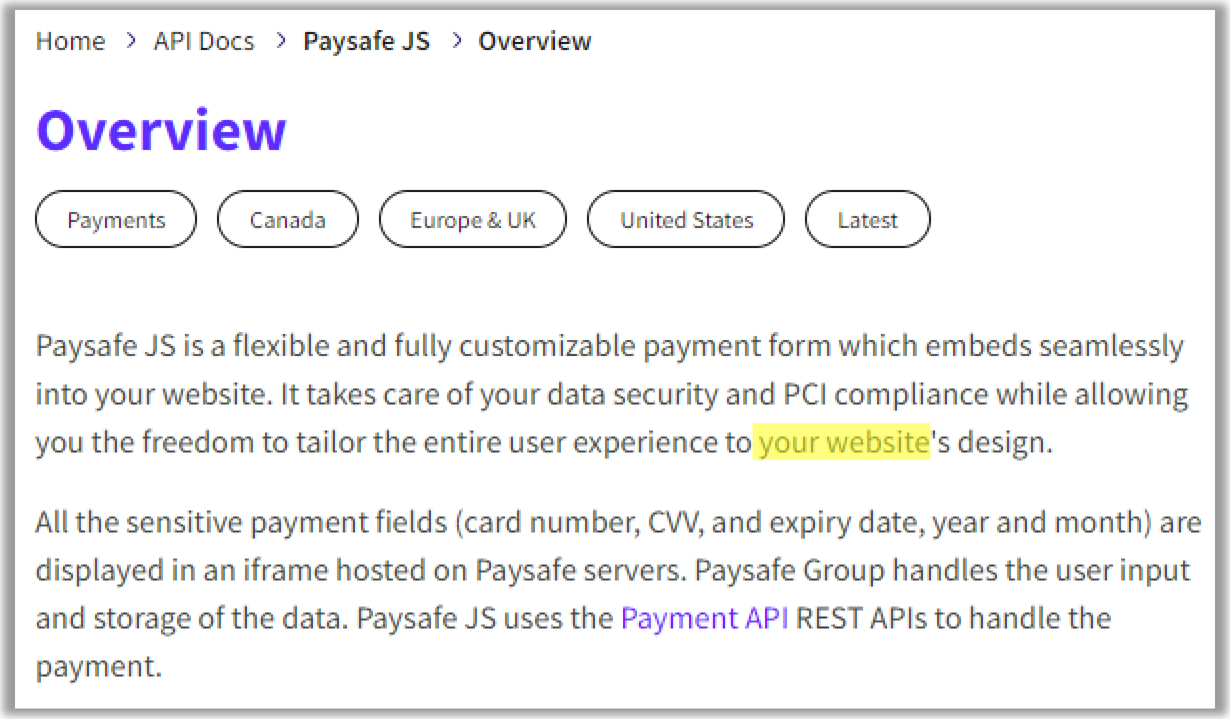
Home > API Docs > Payments API > Overview

Overview

Paysafe Payments API allows you to accept payments via multiple payment methods using a single endpoint integration to process payments from your customers. Connect your application directly to the Payments API to process a full suite of REST-based payment methods.

29. Defendants provide, by the one or more secure servers to a merchant server providing a webpage to a payer computing system used by the payer, an application programming interface (API). This is shown by, *e.g.*, the following excerpt from Defendants' documentation and marketing materials for their "Paysafe JS" and "Payment API" products:[10]

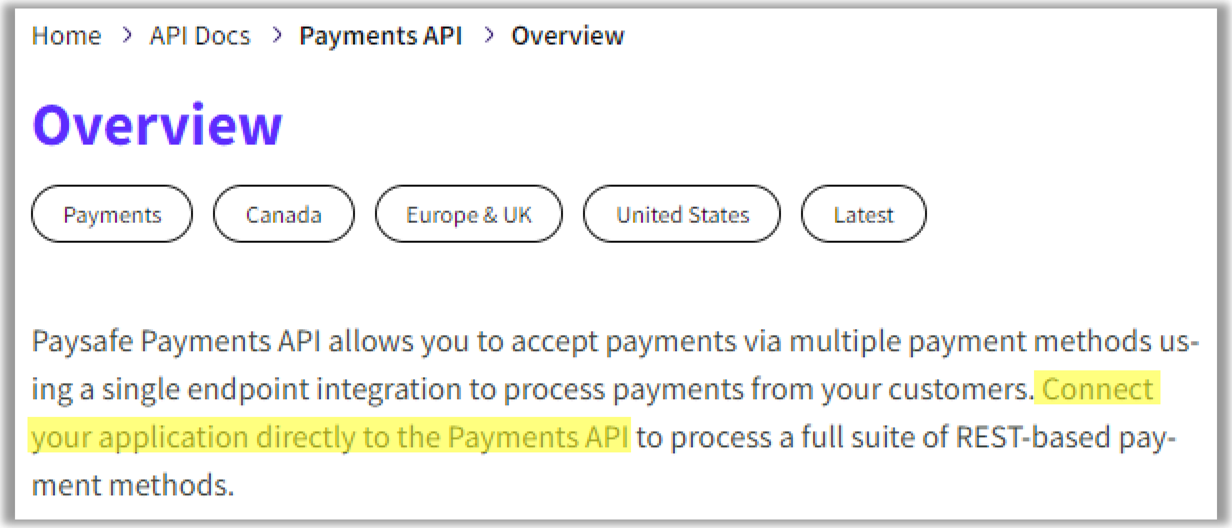
[10] *Id.*; https://www.paysafe.com/en/paysafe-payments-api/ (last visited March 31, 2024).



Overview

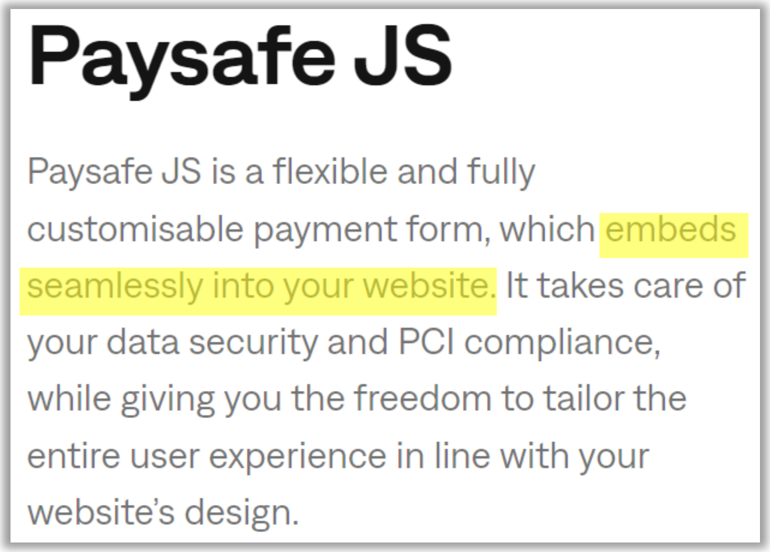
Paysafe JS is a flexible and fully customizable payment form which embeds seamlessly into your website. It takes care of your data security and PCI compliance while allowing you the freedom to tailor the entire user experience to your website's design.

All the sensitive payment fields (card number, CVV, and expiry date, year and month) are displayed in an iframe hosted on Paysafe servers. Paysafe Group handles the user input and storage of the data. Paysafe JS uses the Payment API REST APIs to handle the payment.



Overview

Paysafe Payments API allows you to accept payments via multiple payment methods using a single endpoint integration to process payments from your customers. Connect your application directly to the Payments API to process a full suite of REST-based payment methods.

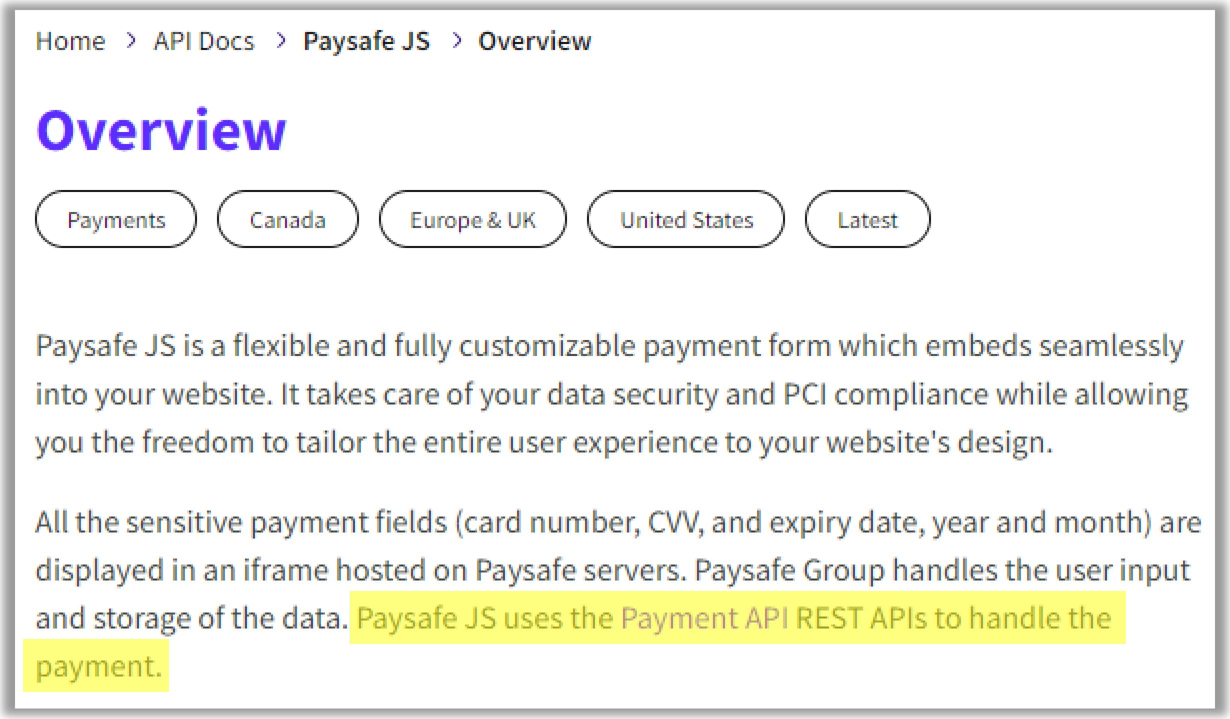


Paysafe JS

Paysafe JS is a flexible and fully customisable payment form, which embeds seamlessly into your website. It takes care of your data security and PCI compliance, while giving you the freedom to tailor the entire user experience in line with your website's design.

30. Defendants' API provides financial account registration and token retrieval functions that can be executed to process the payment transaction. This is shown by, *e.g.*, the following excerpt from Defendants' description of their "Payments API" product in the documentation for their "Paysafe JS" product:[11]

[11] https://developer.paysafe.com/en/api-docs/paysafe-js/overview/ (last visited March 31, 2024)

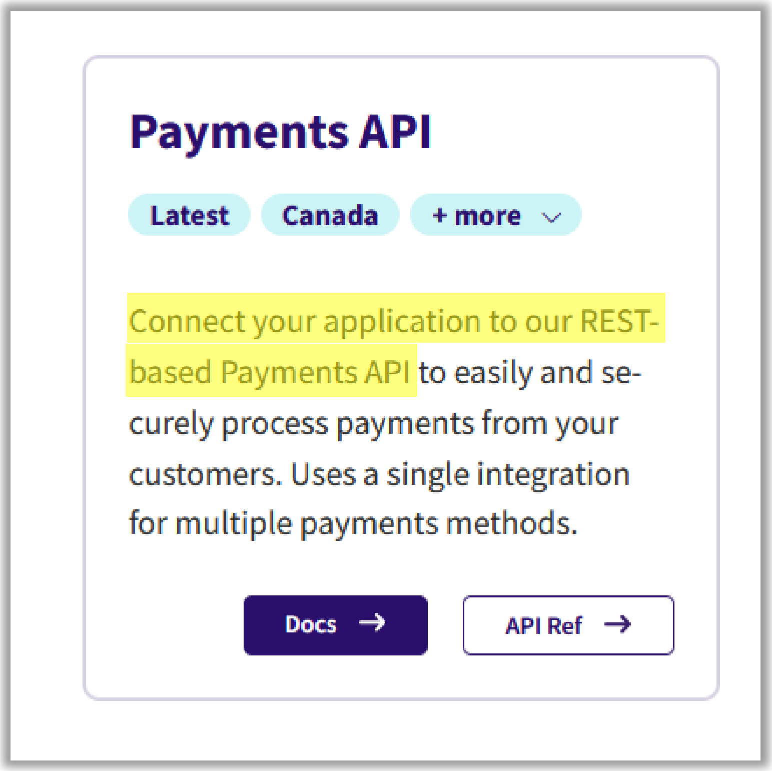


Paysafe JS is a flexible and fully customizable payment form which embeds seamlessly into your website. It takes care of your data security and PCI compliance while allowing you the freedom to tailor the entire user experience to your website's design.

All the sensitive payment fields (card number, CVV, and expiry date, year and month) are displayed in an iframe hosted on Paysafe servers. Paysafe Group handles the user input and storage of the data. Paysafe JS uses the Payment API REST APIs to handle the payment.

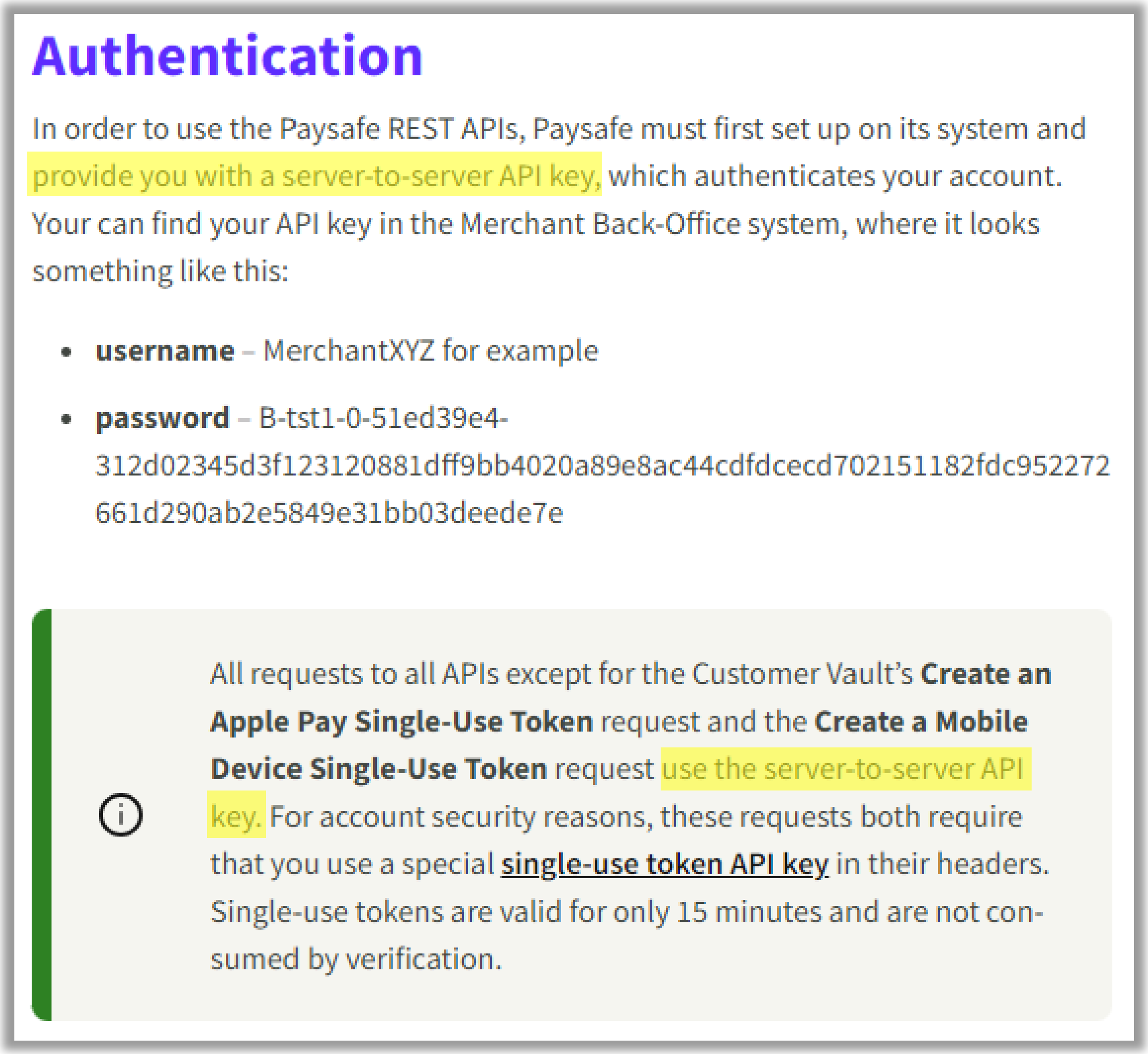
31. Defendants' API provides access to the financial account registration and token retrieval functions to the merchant server. This is shown by, *e.g.*, the following excerpt from Defendants' documentation for their "Payments API" product:[12]

[12] https://developer.paysafe.com/en/api-docs/ (last visited March 31, 2024); https://developer.paysafe.com/en/support/reference-information/authentication/ (last visited March 31, 2024).



Payments API

Connect your application to our REST-based Payments API to easily and securely process payments from your customers. Uses a single integration for multiple payments methods.



Authentication

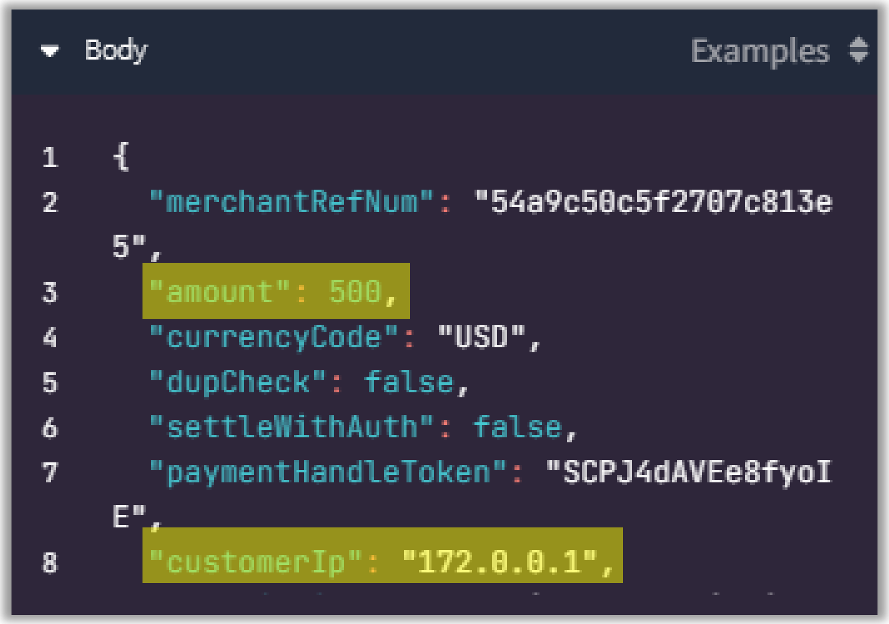
In order to use the Paysafe REST APIs, Paysafe must first set up on its system and provide you with a server-to-server API key, which authenticates your account. Your can find your API key in the Merchant Back-Office system, where it looks something like this:

- **username** – MerchantXYZ for example
- **password** – B-tst1-0-51ed39e4-312d02345d3f123120881dff9bb4020a89e8ac44cdfdcecd702151182fdc952272661d290ab2e5849e31bb03deede7e

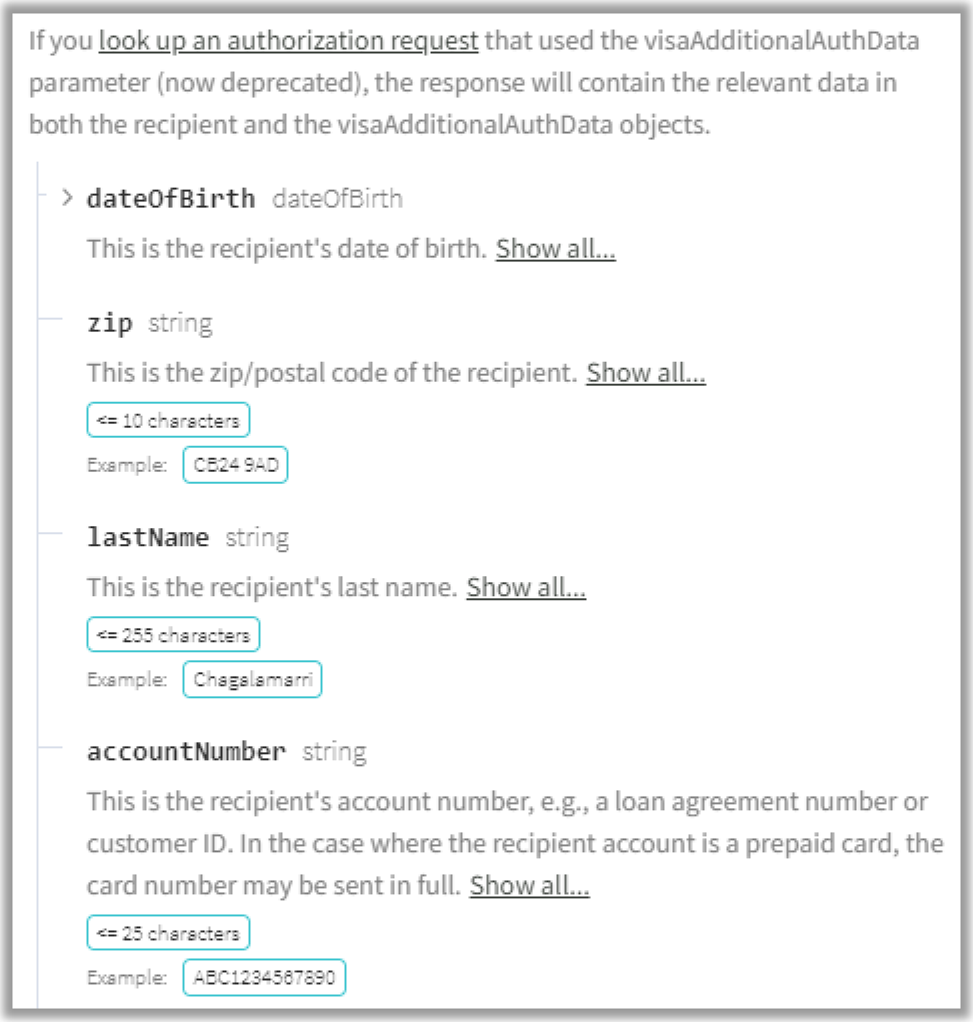
ⓘ All requests to all APIs except for the Customer Vault's **Create an Apple Pay Single-Use Token** request and the **Create a Mobile Device Single-Use Token** request use the server-to-server API key. For account security reasons, these requests both require that you use a special **single-use token API key** in their headers. Single-use tokens are valid for only 15 minutes and are not consumed by verification.

32. Defendants' API receives, from the merchant server via the API, at least one data element associated with the payer and a payment amount from the payer to the payee. This is shown by, *e.g.*, the "customerIp," "dateOfBirth," "zip," "lastName," "accountNumber," and "amount" parameters of the "Process Payment" section of Defendants' documentation for their "Payments API" product:[13]

[13] https://developer.paysafe.com/en/payments-api/#/operations/process-payment (last visited March 31, 2024).



```
{
   "merchantRefNum": "54a9c50c5f2707c813e
5",
   "amount": 500,
   "currencyCode": "USD",
   "dupCheck": false,
   "settleWithAuth": false,
   "paymentHandleToken": "SCPJ4dAVEe8fyoI
E",
   "customerIp": "172.0.0.1",
```



If you look up an authorization request that used the visaAdditionalAuthData parameter (now deprecated), the response will contain the relevant data in both the recipient and the visaAdditionalAuthData objects.

dateOfBirth dateOfBirth

This is the recipient's date of birth. Show all...

zip string

This is the zip/postal code of the recipient. Show all...

<= 10 characters

Example: CB24 9AD

lastName string

This is the recipient's last name. Show all...

<= 255 characters

Example: Chagalamarri

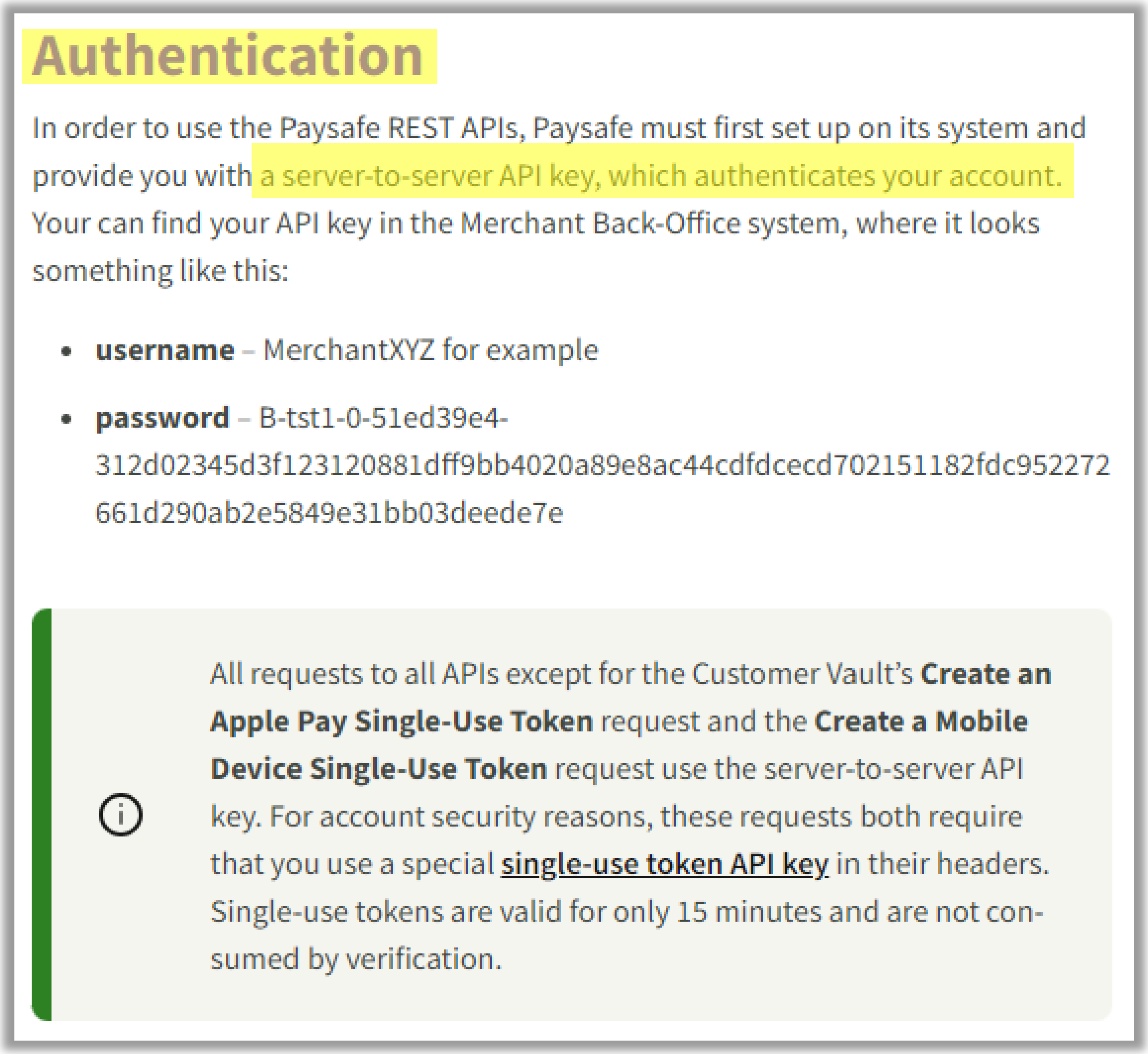
accountNumber string

This is the recipient's account number, e.g., a loan agreement number or customer ID. In the case where the recipient account is a prepaid card, the card number may be sent in full. Show all...

<= 25 characters

Example: ABC1234567890

33. Defendants' API authenticates the payee. This is shown by, *e.g.*, the "API Key" described in Defendants' documentation for their "Payments API" product:[14]



Authentication

In order to use the Paysafe REST APIs, Paysafe must first set up on its system and provide you with a server-to-server API key, which authenticates your account. Your can find your API key in the Merchant Back-Office system, where it looks something like this:

- **username** – MerchantXYZ for example
- **password** – B-tst1-0-51ed39e4-312d02345d3f123120881dff9bb4020a89e8ac44cdfdcecd702151182fdc952272661d290ab2e5849e31bb03deede7e

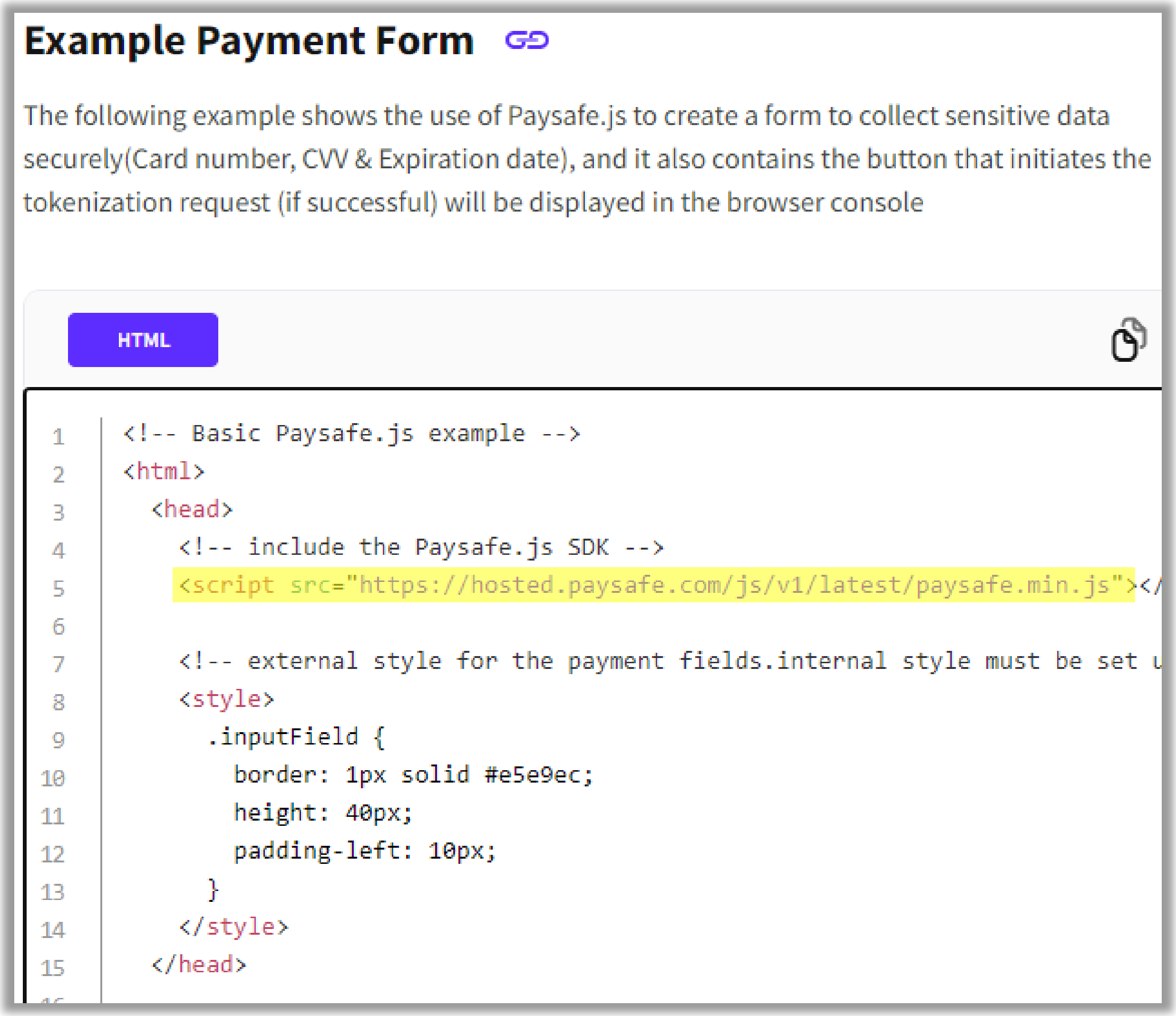
ⓘ All requests to all APIs except for the Customer Vault's **Create an Apple Pay Single-Use Token** request and the **Create a Mobile Device Single-Use Token** request use the server-to-server API key. For account security reasons, these requests both require that you use a special **single-use token API key** in their headers. Single-use tokens are valid for only 15 minutes and are not consumed by verification.

34. Defendants' API executes the financial account registration function, upon initiation by the merchant server, by:

a. (i) Generating a uniform resource locator (URL), for establishing a secure socket layer connection via the internet between the secure server and the payer computing

[14] https://developer.paysafe.com/en/support/reference-information/authentication/ (last visited March 31, 2024).

system, the URL comprising either: a dynamic URL generated by the secure server for the payer and the payee; or a static URL and a hypertext transport protocol (HTTP) parameter used by the secure server to identify the payer and the payee. This is shown by, *e.g.*, URL within "paysafe.min.js" described under "Example Payment Form" Defendants' documentation for their "Paysafe JS" product:[15]



Example Payment Form

The following example shows the use of Paysafe.js to create a form to collect sensitive data securely(Card number, CVV & Expiration date), and it also contains the button that initiates the tokenization request (if successful) will be displayed in the browser console

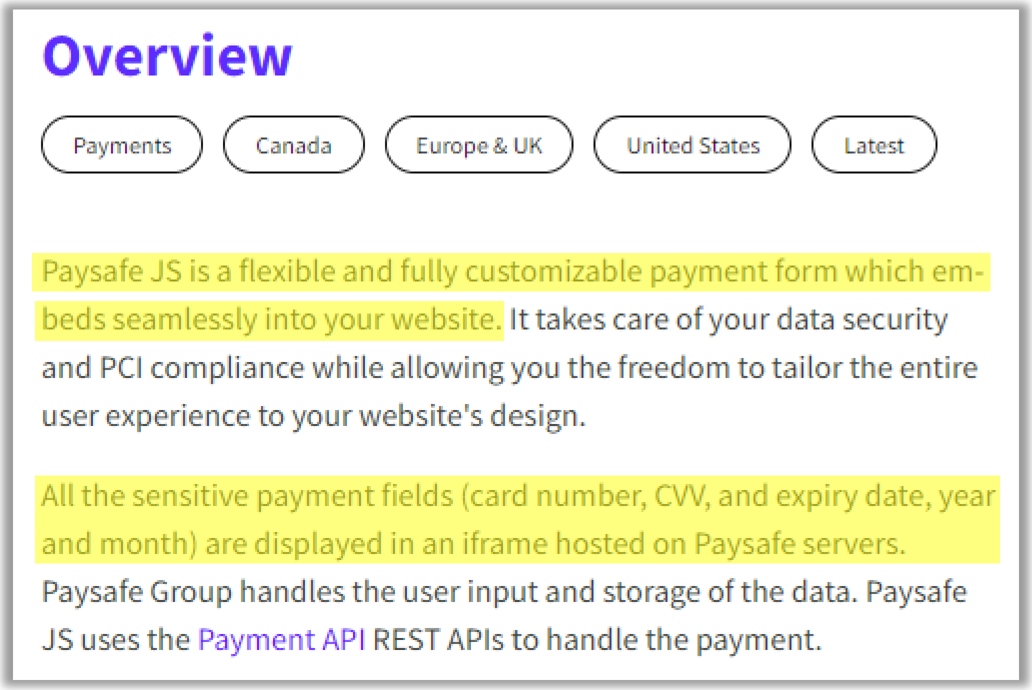
```
<!-- Basic Paysafe.js example -->
<html>
  <head>
    <!-- include the Paysafe.js SDK -->
    <script src="https://hosted.paysafe.com/js/v1/latest/paysafe.min.js"></

    <!-- external style for the payment fields.internal style must be set u
    <style>
      .inputField {
        border: 1px solid #e5e9ec;
        height: 40px;
        padding-left: 10px;
      }
    </style>
  </head>
```

b. (ii) Establishing the secure socket layer connection, in response to an HTTP request received from the merchant server for the generated URL, between the secure

[15] https://developer.paysafe.com/en/api-docs/paysafe-js/overview/ (last visited March 31, 2024).

server and the payer computing system within a window or frame that is displayed within the webpage provided by the merchant server. This is shown by, *e.g.*, the fact that Defendants' products use "HTTPS" URLs, which establish secure socket layer connections.[16]

c. (iii) Outputting instructions to the payer computing system, in response to the HTTP request for the generated URL, to render a financial account registration request form, within the window or frame that is displayed within the webpage provided by the merchant server, that provides functionality for the payer to provide sensitive financial account information associated with a financial account. This is shown by, *e.g.*, Defendants' description of their Paysafe JS product:[17]



Overview

Paysafe JS is a flexible and fully customizable payment form which embeds seamlessly into your website. It takes care of your data security and PCI compliance while allowing you the freedom to tailor the entire user experience to your website's design.

All the sensitive payment fields (card number, CVV, and expiry date, year and month) are displayed in an iframe hosted on Paysafe servers. Paysafe Group handles the user input and storage of the data. Paysafe JS uses the Payment API REST APIs to handle the payment.

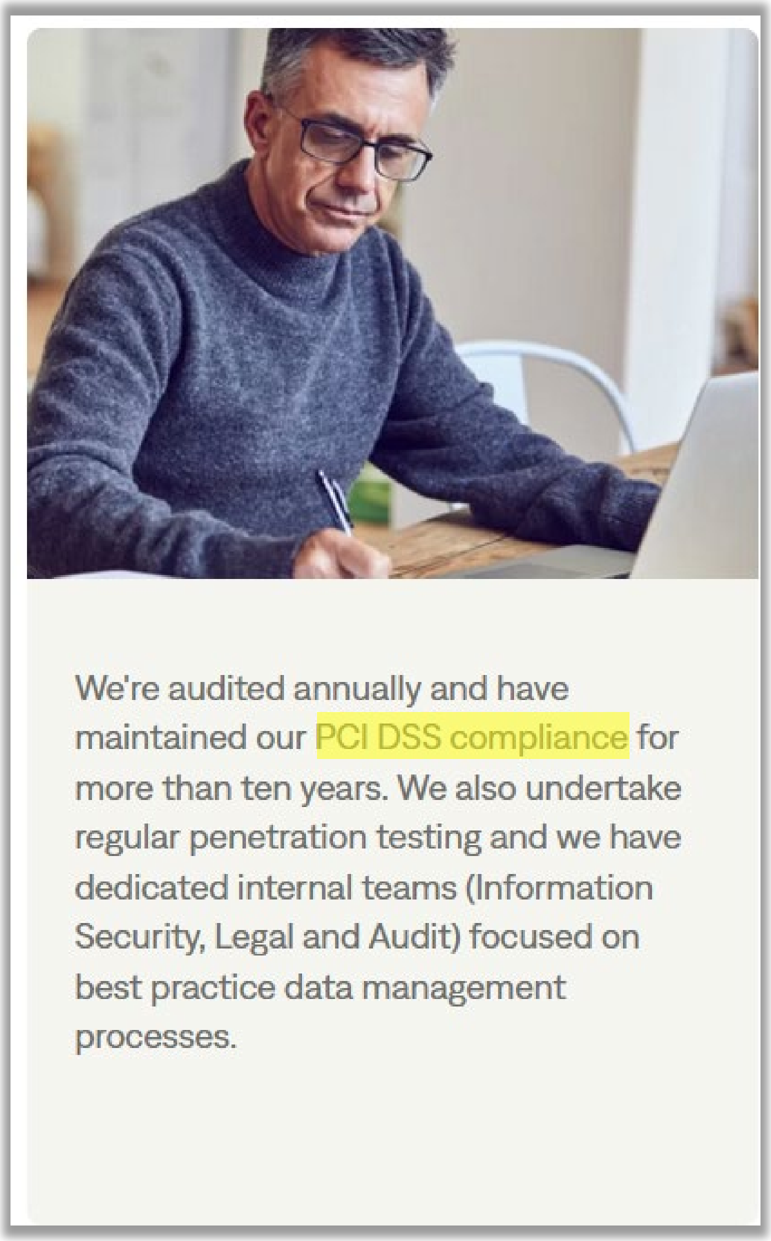
d. (iv) And outputting instructions to the payer computing system, in response to the

[16] *Id.*

[17] https://developer.paysafe.com/en/api-docs/paysafe-js/overview/ (last visited March 31, 2024).

HTTP request for the generated URL, to encrypt the sensitive financial account information provided by the payer and transmit the encrypted financial account information to the secure server via the secure socket layer connection. This is shown by, *e.g.*, the same portions of the documentation discussed in the previous paragraphs and by Defendants' marketing materials regarding their "PCI DSS compliance," which requires transmissions of cardholder data to be encrypted:[18]

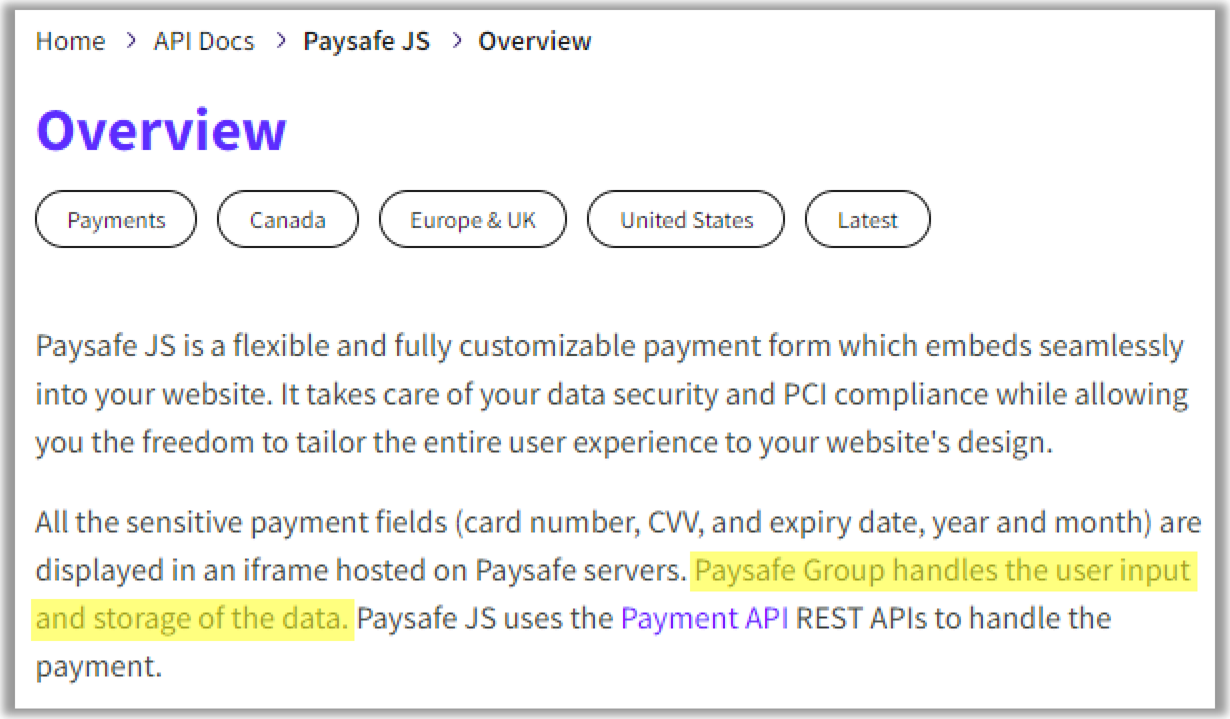
[18] *See id.*; *see also* https://www.paysafe.com/us-en/pci-data-security/ (last visited March 31, 2024).



We're audited annually and have maintained our PCI DSS compliance for more than ten years. We also undertake regular penetration testing and we have dedicated internal teams (Information Security, Legal and Audit) focused on best practice data management processes.

35. Defendants receive the sensitive financial account information provided by the payer via the secure socket layer connection. This is indicated by, *e.g.*, the excerpt shown above and by Defendants' documentation for their "Paysafe JS" product:[19]

[19] https://developer.paysafe.com/en/api-docs/paysafe-js/overview/ (last visited March 31, 2024).



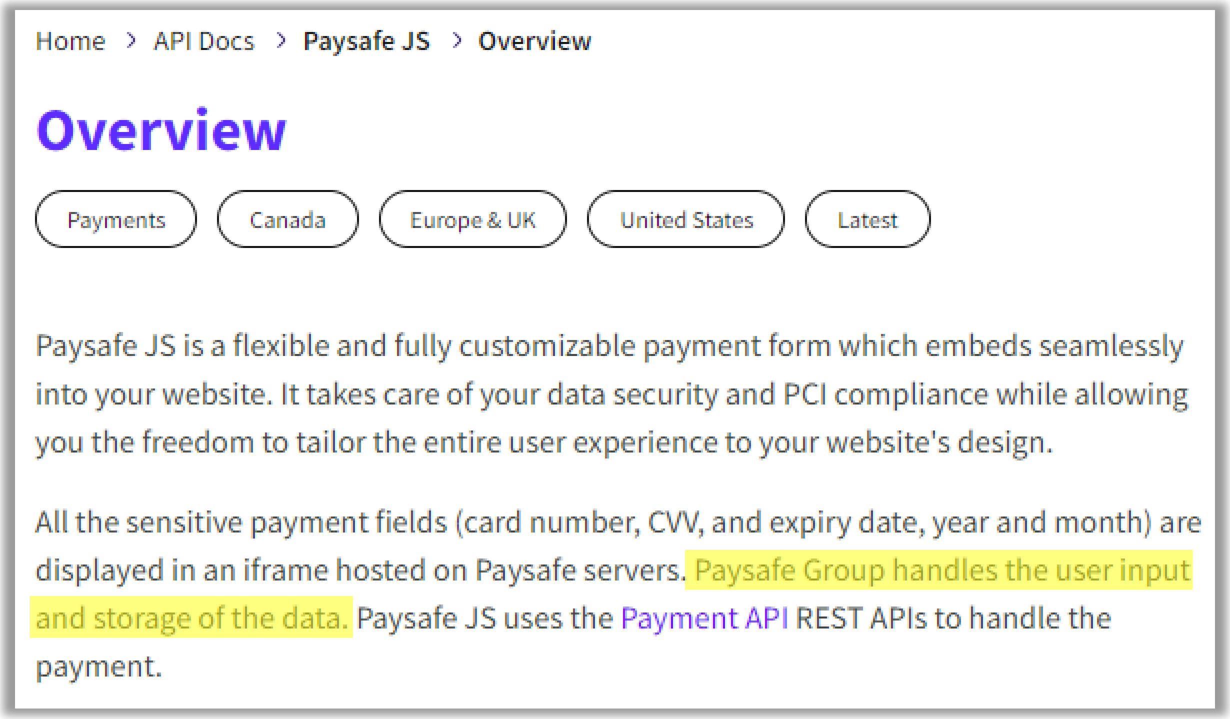
Overview

Paysafe JS is a flexible and fully customizable payment form which embeds seamlessly into your website. It takes care of your data security and PCI compliance while allowing you the freedom to tailor the entire user experience to your website's design.

All the sensitive payment fields (card number, CVV, and expiry date, year and month) are displayed in an iframe hosted on Paysafe servers. Paysafe Group handles the user input and storage of the data. Paysafe JS uses the Payment API REST APIs to handle the payment.

36. Defendants store the sensitive financial account information in a secure storage location and performs each software process required to maintain compliance with one or more information security standards. This is shown by, *e.g.*, the excerpt above and the "customer vault" in the following excerpt from one of Defendants' marketing videos for their "Payment API" product:[20]

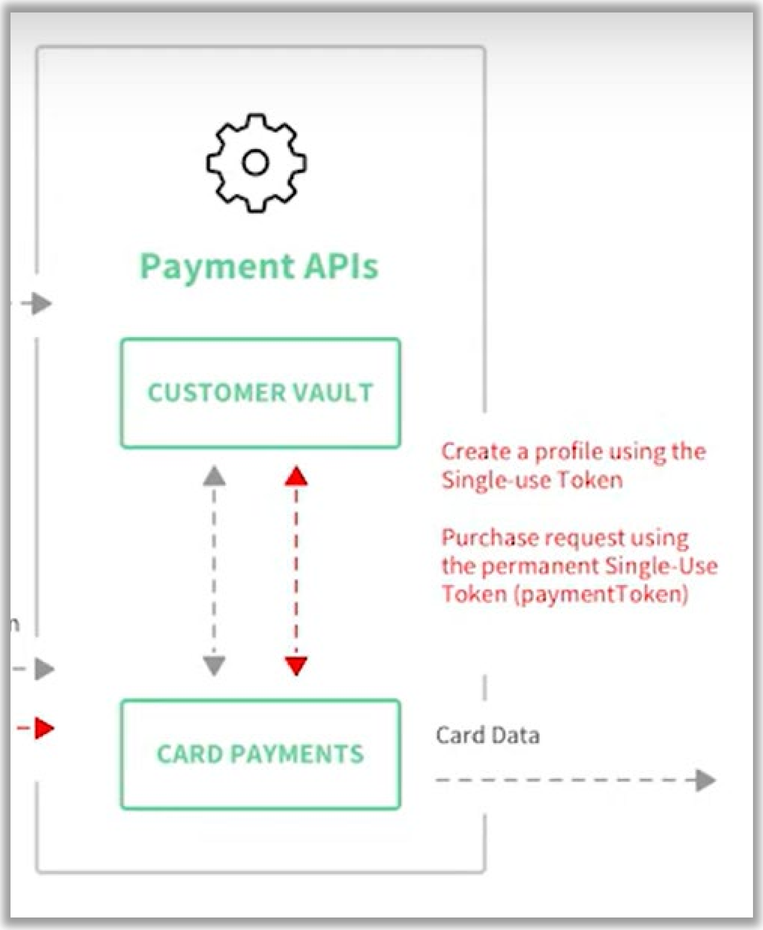
[20] *Id.*; https://www.youtube.com/watch?v=3bDaL9RVYZ8&t=196s (last visited March 31, 2024).



Overview

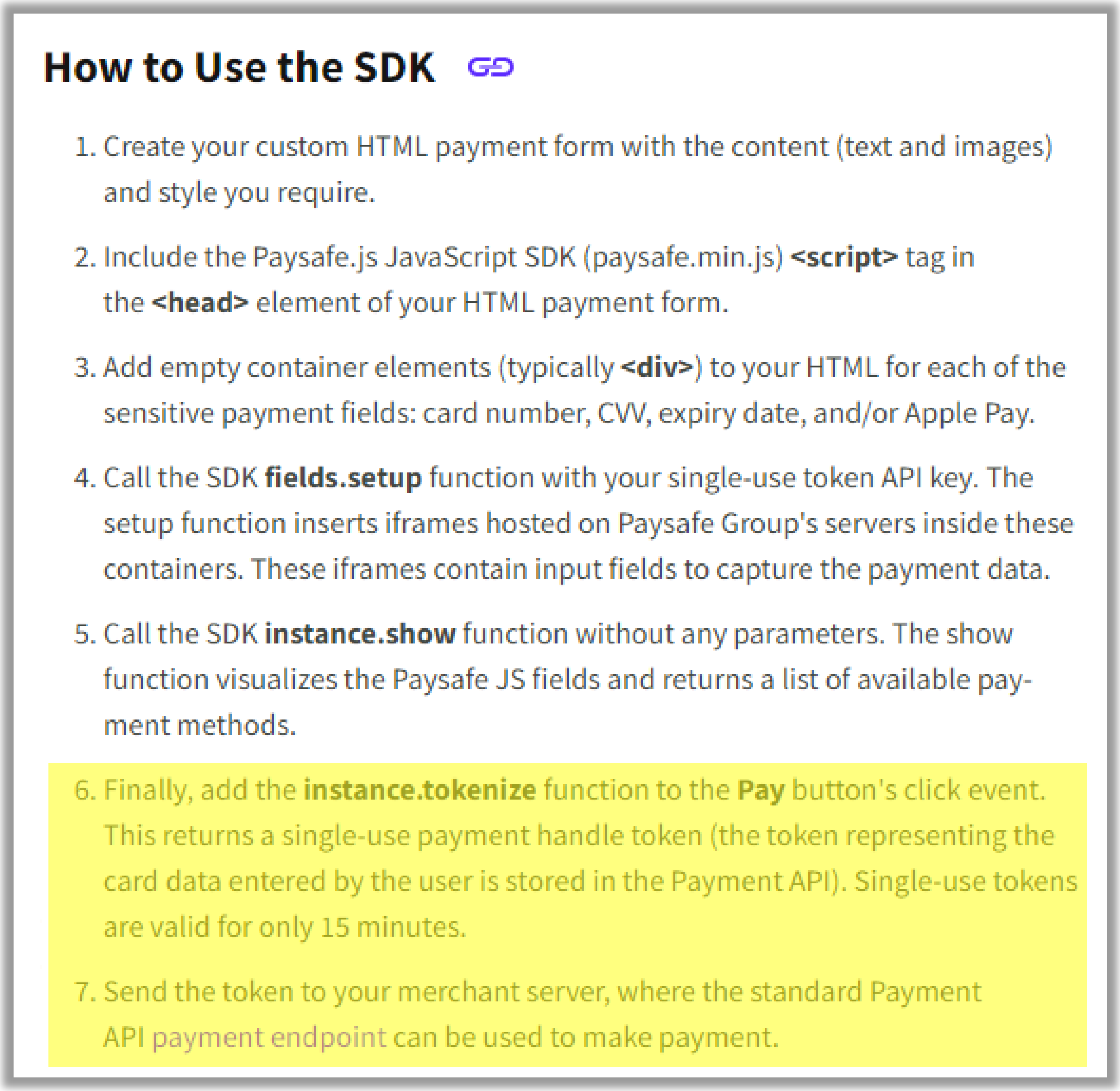
Paysafe JS is a flexible and fully customizable payment form which embeds seamlessly into your website. It takes care of your data security and PCI compliance while allowing you the freedom to tailor the entire user experience to your website's design.

All the sensitive payment fields (card number, CVV, and expiry date, year and month) are displayed in an iframe hosted on Paysafe servers. Paysafe Group handles the user input and storage of the data. Paysafe JS uses the Payment API REST APIs to handle the payment.


Payment APIs
CUSTOMER VAULT
Create a profile using the Single-use Token
Purchase request using the permanent Single-Use Token (paymentToken)
CARD PAYMENTS
Card Data

37. Defendants execute a token retrieval function, upon initiation by the merchant server via the API, by: providing a non-sensitive electronic data token representing the sensitive financial account information to the merchant server without providing the sensitive financial account information to the merchant server and without providing the non-sensitive electronic data token to the payer; and processing the payment transaction using the sensitive financial account information by generating and transmitting an electronic request requesting the payment amount from the financial account, obtaining the payment amount, and forwarding at least a portion of the payment amount to the payee. For example, the "tokens" described in the excerpt below show "a non-sensitive electronic data token," and paragraph 7 of the excerpt below shows that it processes

a payment:[21]



How to Use the SDK

1. Create your custom HTML payment form with the content (text and images) and style you require.
2. Include the Paysafe.js JavaScript SDK (paysafe.min.js) **<script>** tag in the **<head>** element of your HTML payment form.
3. Add empty container elements (typically **<div>**) to your HTML for each of the sensitive payment fields: card number, CVV, expiry date, and/or Apple Pay.
4. Call the SDK **fields.setup** function with your single-use token API key. The setup function inserts iframes hosted on Paysafe Group's servers inside these containers. These iframes contain input fields to capture the payment data.
5. Call the SDK **instance.show** function without any parameters. The show function visualizes the Paysafe JS fields and returns a list of available payment methods.
6. Finally, add the **instance.tokenize** function to the **Pay** button's click event. This returns a single-use payment handle token (the token representing the card data entered by the user is stored in the Payment API). Single-use tokens are valid for only 15 minutes.
7. Send the token to your merchant server, where the standard Payment API payment endpoint can be used to make payment.

38. Defendants had actual knowledge of the Asserted Patent and the infringement of the same no later than the date of this Complaint.

39. Defendants have and continue to indirectly infringe one or more claims of the Asserted Patent by inducing and/or contributing to direct infringement of the Asserted Patent by

[21] https://developer.paysafe.com/en/api-docs/paysafe-js/overview/ (last visited March 31, 2024).

customers, importers, sellers, resellers, and users of the Infringing Methods.

40. Defendants have and continue to induce others to directly infringe, either literally or under the doctrine of equivalents, by, among other things, making, using, offering to sell, selling and/or importing into the United States, without authority, products or services that practice one or more claims of the Asserted Patent.

41. Defendants induced the infringement by others with the intent to cause infringing acts by others or, in the alternative, with the belief that there was a high probability that others infringe the Asserted Patent, but while at best, remaining willfully blind to the infringement.

42. As discussed in Paragraphs 23–37, above, Defendants advertise the Infringing Methods, publish specifications and promotional literature encouraging customers to implement and incorporate the Infringing Methods into end user products, create and/or distribute user manuals for the Infringing Methods that provide instructions and/or encourage infringing use, and offer support and/or technical assistance to their customers that provide instructions on and/or encourage infringing use.

43. Defendants encourage and facilitate their customers to infringe the Asserted Patent by promoting the Infringing Methods, for example, providing documentation with listings of "Advantages" and integration instructions for the "Paysafe JS" product.[22]

44. Defendants' customers that incorporate the Infringing Methods into other products and services (*e.g.*, Genscape, Waterborne Energy, DraftDemons.com, and HogWildPoker) each directly infringe the Asserted Patent pursuant to Defendants' instructions and advertisements.

45. Additionally, Defendants have and continue to contribute to the direct infringement of others, either literally or under the doctrine of equivalents, by, among other things, offering to

[22] https://developer.paysafe.com/en/api-docs/paysafe-js/overview/ (last visited March 31, 2024).

sell or selling within the within the United States, components of a patented device or an apparatus for use in practicing the claimed method, constituting a material part of the invention.

46. As discussed in Paragraphs 23–37, above, Defendants provide APIs and example code for the Infringing Methods that constitute a component of a patented device or an apparatus for use in practicing the claimed method.

47. Defendants do this knowing the same to be especially made or especially adapted for use in an infringement of such patent, and not a staple article or commodity of commerce suitable for substantial noninfringing use.

48. Defendants' customers that incorporate the APIs and example code into other products and services each directly infringe the Asserted Patent.

V. JURY DEMAND

49. Autoscribe hereby demands a trial by jury on all issues so triable.

VI. PRAYER FOR RELIEF

WHEREFORE, Autoscribe requests entry of judgment in its favor and against Defendants as follows:

a) A declaration that Defendants have directly infringed one or more claims of the Asserted Patent, either literally or under the doctrine of equivalents;

b) A declaration that Defendants have induced and/or contributed to infringement and/or are inducing and/or contributing to infringement of one or more claims of the Asserted Patent, either literally or under the doctrine of equivalents;

c) An award of damages pursuant to 35 U.S.C. § 284 adequate to compensate Autoscribe for Defendants' infringement of the Asserted Patent in an amount according to proof at trial (together with prejudgment and post-judgment interest), but no less than a reasonable royalty;

d) An award of costs and expenses pursuant to 35 U.S.C. § 284 or as otherwise permitted by law; and

e) Such other and further relief, whether legal, equitable, or otherwise, to which Autoscribe may be entitled or which this Court may order.

Dated: April 11, 2024

Respectfully submitted,

/s/ Jason McManis

Jason McManis (attorney-in-charge)
Texas Bar No.: 24088032
S.D. Tex. No.: 3138185
Colin Phillips
Texas Bar No.: 24105937
S.D. Tex. No.: 3576569
Chun Deng
Texas Bar No.: 24133178
S.D. Tex. No.: 3860688
Angela Peterson
Texas Bar No.: 24137111
S.D. Tex. No.: 3862849
AHMAD, ZAVITSANOS & MENSING, PLLC
1221 McKinney Street, Suite 2500
Houston, Texas 77010
Tel.: (713) 655-1101
Facsimile: (713) 655-0062
jmcmanis@azalaw.com
cphillips@azalaw.com
cdeng@azalaw.com
apeterson@azalaw.com

Attorneys for Autoscribe Corporation